



Universidad
Carlos III de Madrid



This is a postprint version of the following published document:

Iglesias, J. A.; Ledezma, A.; Sanchis, A. (2013). "Ensemble method based on individual evolving classifiers". *Evolving and Adaptive Intelligent Systems (EAIS)*, 2013 IEEE Conference on, Singapore, 16-19 April 2013, pp. 56-61.

DOI: 10.1109/EAIS.2013.6604105

© 2013 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Ensemble Method Based on Individual Evolving Classifiers

José Antonio Iglesias
Carlos III University of Madrid
Madrid, Spain
Email: jiglesia@inf.uc3m.es

Agapito Ledezma
Carlos III University of Madrid
Madrid, Spain
Email: ledezma@inf.uc3m.es

Araceli Sanchis
Carlos III University of Madrid
Madrid, Spain
Email: masm@inf.uc3m.es.

Abstract—Humans often seek a second or third opinion about an important matter. Then, a final decision is reached after weighing and combining these opinions. This idea is the base of the ensemble based systems. Ensembles of classifiers are well established as a method for obtaining highly accurate classifiers by combining less accurate ones. On the other hand, evolving classifiers are inspired by the idea of evolve their structure in order to adapt to the changes of the environment.

In this paper, we present a proof-of-concept method for constructing an ensemble system based on Evolving Fuzzy Systems. The main contribution of this approach is that the base-classifiers are self-developing (evolving) Fuzzy-rule-based (FRB) classifiers. Thus, we present an ensemble system which is based on evolving classifiers and keeps the properties of the evolving approach classification of streaming data. It is important to clarify that the evolving classifiers are gradually developing but they are not genetic or evolutionary.

I. INTRODUCTION

To make a good decision about an important matter, humans often seek a second opinion, a third opinion, or even any more. Then, these opinions are usually weighted and combined in order to reach a final decision. This process improves our confidence that we are making the right decision. The idea underlying this process is the base of the ensemble based systems. These systems are also known as multiple classifier systems, committee of classifiers or mixture of experts [1].

An ensemble of classifiers is defined by Ditterich [2] as a set of classifiers whose individual decisions are combined in some way (typically by weighted or unweighted voting) to classify new examples. The strategy in ensemble systems is therefore to create a collection of individual accurate classifiers, and combine their outputs such that the combination improves upon the performance of a single classifier. This requires, however, that individual classifiers make errors on different instances, that is, that ensemble members are diverse.

There are many different researches which propose different methods for constructing good ensembles of classifiers. The researches in this field come to the conclusion that ensembles are often much more accurate than the individual classifiers that make them up. According to Ditterich [2], ensembles can improve performance because uncorrelated errors made

by individual classifiers can be removed by voting. However, there are still many questions about the best way to construct ensembles as well as issues about how best understand the decisions made by ensembles.

Two of the most well known methods to construct ensembles of classifiers are: Bagging [3] and Boosting [4]. Bagging builds sets of data of the same size of the original data set by applying random sampling with replacement. Boosting also resamples original data set with replacement, but the training set used for each member of the ensemble is chosen based on the performance of the earlier classifier(s) in the set. That is, in Boosting, examples that are incorrectly predicted by the current ensemble are chosen more often than examples that were correctly predicted.

In this paper, we present a proof-of-concept ensemble method in which all the individual classifiers are Evolving Fuzzy-rule-based Classifier (*eClass*).

An *eClass* is a fuzzy rule-based classifier with rules that are evolved from streaming data. Thus, the use of an *eClass* does not need to know beforehand in how many classes the data will be classified as new classes can be introduced during the learning process [5]. An *eClass* learns new rules from new data gradually preserving/inheriting the rules learned already. Thus, *eClass* can be defined as a self-developing classifier which has both their parameters but also (more importantly) their structure self-adapting on-line. In addition, an *eClass* can start learning "from scratch".

It is important to clarify that an *eClass* is not a genetic or evolutionary classifier. A genetic/evolutionary FRB classifier [6], [7] is a set of input/output rules that are modified by a genetic algorithm. In that case, new rules are generated as a crossover or mutation of previous rules. However, an *eClass* learns new rules from new data gradually preserving the rules already learned.

eClass has been applied to a wide range of problems, both benchmarks and real. The main contribution of this paper is the use of evolving classifiers in an ensemble system. Although there are several evolutionary classifier ensembles

[8], [9]; to the best of our knowledge, we are the first to propose and investigate ensemble methods based on evolving fuzzy classifiers.

This paper is organized as follows: Section 2 provides an overview of the background and related work of the two main concepts used this research: ensemble of classifiers and evolving fuzzy classifiers. The overall structure of the proposed ensemble system is described in detail in Section 3. Section 4 describes the experimental settings and the experimental results obtained. Finally, Section 5 contains concluding remarks and future work.

II. BACKGROUND AND RELATED WORK

As we propose an ensemble method based on evolving classifiers, we will define the background and related work of the two main concepts used this research: ensemble of classifiers and evolving fuzzy classifiers.

A. Ensemble of Classifiers

One of the first works on ensemble systems was proposed in 1979 [10]. In this early work, the authors propose the partition of the feature space using two or more classifiers. In 1990, a new research [11] concluded that the generalization performance of a neural network can be improved using an ensemble of neural networks. Bagging [3] and Boosting [4] are two of the most well-known ensemble learning methods due to their theoretical performance guarantees and strong experimental results.

The literature of ensemble classifiers is truly vast. There are many different approaches which propose different methods for generating individual classifiers, and strategies for combining the outputs of these classifiers. Some of these approaches were defined as combination of multiple classifiers [12][13], dynamic classifier selection [14], classifier fusion [15], classifier ensembles [16] and many others.

There are many different areas in which ensemble of classifiers have been used, and these areas are continuously growing. As it is described by Polikar [1], some of the more promising areas include using ensemble systems in non-stationary environments [16] or in clustering applications [17], [18]. Some of the practical applications in which these ensemble systems are rapidly growing are: biomedical [19], financial [20], remote sensing [21], or chemical [22].

B. Evolving Fuzzy Classifiers: *eClass*

eClass (evolving Classifier) family [5] is a set of evolving neuro-fuzzy classifiers which take its roots in evolving Takagi-Sugeno (eTS). The first evolving fuzzy rule-based systems are introduced in [23] and further developed in [24].

During training in these classifiers, a set of fuzzy rules that describes the most important features for the classification

of each class is formed. These rules are constantly adjusted to the available training data. One of the advantages of *eClass* is that it does not require parameter optimization as its only parameter 'scale' can be directly inferred from the training data. This technique [25] is based on partitioning the data space into overlapping local regions through *Recursive Density Estimation* (RDE) and associating clusters (respectively fuzzy sets) to them.

As it is explained in [5], the main differences between *eClass* family and a conventional Fuzzy Rule-Based (FRB) classifier are:

- the open structure of the rule-base: *eClass* self-develops on-line starting from scratch, while in a conventional FRB classifier it is determined offline and then fixed.
- the online learning mechanism which takes into account this flexible rule-base structure.

eClass family includes two different architectures and on-line learning methods:

- *eClass0* with the classifier consequents representing class label.
- *eClass1* for regression over the features using first order eTS fuzzy classifier.

eClass family has been applied in many different areas with a great success. An important area of application of these evolving classifiers is robotics and autonomous systems. Some examples of application include: autonomous landmark recognition [26], self-localization and mapping [27], object detection and tracking [28][29], collision avoidance [30], activity recognition from sensor streams [31][32] and user modeling [33][34]. In this research, *eClass0* will be applied as base-classifier in the proposed ensemble.

C. Reasons for using ensemble of evolving classifiers

According to Polikar [1], there are several theoretical reasons why an ensemble system is preferred:

- 1) *Statistical Reasons*: A set of classifiers with similar training performances may have different generalization performances.
- 2) *Large Volumes of Data*: Training a classifier with vast amount of data is usually not practical. Training different classifiers with different partitions of data and combining their outputs proves to be a more efficient approach.
- 3) *Too Little Data*: In the absence of adequate training data, resampling techniques can be used for drawing overlapping random subsets of the available data.
- 4) *Divide and Conquer*: The complex decision boundary could be approximated by an appropriate combination of different classifiers.

5) *Data Fusion*: Ensemble based approaches have successfully been used for applications in which data from different sources are combined.

In the method that we propose in the next section, as we are using evolving classifiers, they can be generated on-line from streaming data. As evolving classifiers can cope with huge amounts of data, the use of large volumes of data or little data is not a reason for using ensembles in our case. However, the other reasons (especially, statistical and “*divide an conquer*” reasons) proposed by Polikar are quite interesting and we would like to check if using evolving classifiers these reasons are also true.

III. ENSEMBLE METHOD BASED ON EVOLVING CLASSIFIERS

This section introduces the proposed ensemble method based on evolving classifiers, which will be named as *eEnsemble*. The section is divided in two parts: First, we will detail *eClass0*, the evolving classifier used for the individual classifiers. Then, the architecture of the proposed ensemble method will be explained in detail.

A. Evolving Classifier: *eClass0*

eClass0 possesses a zero-order Takagi-Sugeno consequent, so a fuzzy rule in the *eClass0* model has the following structure:

$$\text{Rule}_i = \text{IF}(X_1 \text{ is } P_1) \text{ AND } \dots \text{AND } (X_n \text{ is } P_n) \\ \text{THEN } \text{Class} = \text{Class}_i \quad (1)$$

where i represents the number of rule; n is the number of input variables; the vector X stores the input variables and the vector P stores the values of the features of one of the prototypes (cluster centre) of the corresponding class. $\text{Class} \in \{\text{set of different classes}\}$.

The *eClass0* model is composed of several fuzzy rules per class (the number of rules depends on the heterogeneity of the input data of the same class). During the training process, a set of rules is formed from scratch using an evolving clustering approach to decide when to create new rules. The inference in *eClass0* is produced using the “winner takes all” rule and the membership functions that describe the degree of association with a specific prototype are of Gaussian form. The *potential* (Cauchy function of the sum of distances between a certain data sample and *all* other data samples in the feature space) is used in the partitioning algorithm. However, in these classifiers, the potential (P) is calculated recursively (which makes the algorithm faster and more efficient). The potential of the k^{th} data sample (x_k) is calculated [5] by the equation 2. The result of this function represents the *density* of the data that surrounds a certain data sample.

$$P(x_k) = \frac{1}{1 + \frac{\sum_{i=1}^{k-1} \text{distance}(x_k, x_i)}{k-1}} \quad (2)$$

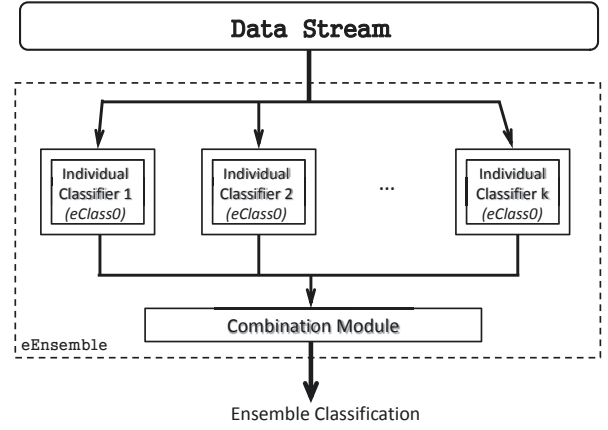


Fig. 1: *eEnsemble* Architecture.

where *distance* represents the distance between two samples in the data space.

The potential can be calculated using the euclidean or the cosine distance. In this case, cosine distance (*cosDist*) is used to measure the similarity between two samples; as it is described in equation 3.

$$\text{cosDist}(x_k, x_p) = 1 - \frac{\sum_{j=1}^n x_{kj} x_{pj}}{\sqrt{\sum_{j=1}^n x_{kj}^2 \sum_{j=1}^n x_{pj}^2}} \quad (3)$$

where x_k and x_p represent the two samples to measure its distance and n represents the number of different attributes in both samples.

Note that the expression in the equation 2 requires all the accumulated data sample available to be calculated, which contradicts to the requirement for real-time and on-line application needed in the proposed problem. For this reason, in [5] it is developed a recursive expression cosine distance.

All details about the *eClass0* model and the learning algorithm can be found in [25].

B. Architecture of the proposed method: *eEnsemble*

The ensemble that we propose in this paper is composed of a number of more simple classifiers (individual or base-classifiers). The number of these individual classifiers (k) must be defined at the beginning by hand. All of these k classifiers have the same *eClass0* structure, but they will be trained with different (and disjoint) sets of data.

The architecture of the proposed method is shown in Figure 1 and it can be divided in three steps: Receiving the data streams, creating the individual classifiers and combining individual classifiers. These parts are explained in the following subsections:

1) *Receiving the data streams*: Unlike most of the ensembles of classifiers, the method that we propose has the following characteristic:

- The classification is online,
- The inputs of the method are streaming data.

As streaming data arrive continuously, there is not an initial data set. In this case, when a new sample arrives, it is sent randomly to one of the k individual classifiers. This on-line partitioning is done without taking into account any variable or technique; it is done entirely randomly. Thus, the examples are randomly distributed across the set of individual classifiers.

2) *Creating the Individual Classifiers*: Once an example is received by an individual classifier, it is analyzed using *eClass0*. As we already explained before, this classifier is designed with an evolving (self-developing) FRB structure. Also, this classifier can start "from scratch" and it is no need to define any variable or parameter for the different classifiers.

It is important to highlight that the (flexible) rule-base structure of the individual classifiers will be different since the data streams that they receive are also different. Thus, the number of rules and the focal points of the fuzzy rules will be different in the individual classifiers. This is the main characteristic of the proposed method.

3) *Combining Individual Classifiers (Combination Module)*: Once the individual classifiers have been trained with example data streams, we can use the proposed *eEnsemble* in order to classify a new example/instance. In this case, the new example will be tested by all the classifiers and their output will be combined in order to obtain a final classification. This task can be achieved by using different methods: Linear functions like average function [35]; non-linear combination methods, like majority voting (Bagging, Boosting) or meta-learning methods [36] [37].

In the ensemble that we propose in this paper, we will take the simplest approach: unweighted vote. The outputs of the individual classifiers will be combined by taking a majority vote of their classification. For any given instance, the class chosen by most individual classifiers is the ensemble classification. This method is also used in Bagging [3].

Although this is a very simple method, the experience in the forecasting literature has been that simple, unweighted voting is robust [38]. However, this module could be changed without changing the base idea of the proposed ensemble.

C. *eEnsemble variaton*: *eEnsembleR*

One of the characteristic of Bagging is that it creates the ensembles by repeatedly randomly resampling the training data. However, in *eEnsemble*, an instance will be only the input of a specific base-classifier (randomly chosen). For this reason, in order to take into account that an example could be

the input of several base-classifiers, we propose a variation named *eEnsembleR*.

The only difference between *eEnsemble* and *eEnsembleR* is that in this last method, when an instance arrives, it will go (randomly) to one of the base-classifiers or to all of them. In this case, each instance has the probability of 1/2 to be repeated in all of the base-classifiers. This version can give us an idea about how the addition of the same instance in different individual classifiers can vary the result of the proposed method.

IV. EXPERIMENTAL SETUP AND RESULTS

The two proposed methods (*eEnsemble* and *eEnsembleR*) have been tested on some benchmark problems from the UCI machine learning repository [39]. To apply *eEnsemble* to these data sets, we consider them as pseudo-online streams. Thus, although we are using data sets in an offline mode, the proposed method is designed to be used with data streams. The reason for this experiments is to achieve comparable results and obtain further validation of the proposed ensembles.

A. *Benchmark Problems from UCI Repository*

The different datasets used in this experimentation are detailed as follows:

1) *Wine Data Set*: These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the 3 types of wines.

2) *Cardiotocography (CGT) Data Set*: This dataset contains 2126 fetal cardiotocograms (CTGs) which were automatically processed and the respective diagnostic features measured.

3) *Waveform Database Generator (VI) Data Set*: This dataset describes 3 classes of waves which are described by 21 attributes and all of the instances include noise.

4) *Page Blocks Data Set*: This is a set of 5473 examples which come from 54 distinct documents. Each observation concerns one block.

5) *Statlog (Landsat Satellite) Data Set*: This dataset consists of the multi-spectral values of pixels in 3x3 neighborhoods in a satellite image, and the classification associated with the central pixel in each neighbourhood.

6) *Pen-Based Recognition of Handwritten Digits Data Set*: This data set contains the handwriting information from a pressure sensitive tablet with an integrated LCD display. This information is classified in ten digits (from 0 to 9).

7) *Statlog (Shuttle) Data Set*: This data set contains 58000 instances (with 9 attributes) which are classified in 8 different classes.

The data set size and other characteristic of the data sets are listed in Table I. Note that the number of instances is very different in each dataset.

TABLE I: Description of the Datasets.

| DataSet | Number of Instances | Number of Attributes | Number of Classes |
|------------------|---------------------|----------------------|-------------------|
| Wine | 178 | 13 | 3 |
| Cardiotocography | 2126 | 23 | 11 |
| Waveform | 5000 | 21 | 3 |
| Page Blocks | 5473 | 10 | 6 |
| Landsat | 6435 | 36 | 8 |
| Pen Digit | 10992 | 16 | 10 |
| Shuttle | 58000 | 9 | 8 |

B. Experimental Setup

As we already mentioned, before applying *eEnsemble*, it is necessary to parameterize the number of individual classifiers (represented as k) that will make up the ensemble. As the result of the *eEnsemble* varies according to this parameter, we have used there different values for k : 3, 5 and 10.

The performance of the ensemble that we propose depends on how the data stream is distributed among the individual classifiers. As this distribution is randomly done, the result of the *eEnsemble* can change because of this aspect. For this reason, the results shown in this section are the average of 100 different executions of *eEnsemble*.

Finally, in order to evaluate the performance of *eEnsemble*, it is compared with the *eClass0*. Also, to measure the performance of the different methods, for each data set, 80 percent of the data are kept as training instances while the rest are used as the pool of test examples.

The ensemble *eEnsembleR* has been evaluated with the same experimental setup described previously.

C. Experimental Results

Table II shows the rate for *eClass0* and the average rate for *eEnsemble* and *eEnsembleR* in any of the 7 different datasets. Major rows correspond to the 7 different datasets and columns show the results of the experiment for each model (and also for the number of individual classifiers (k) in the case of the proposed ensembles). According to the results, we can conclude that, in general, the results of the proposed ensembles are comparable with those obtained from *eClass0*. Nevertheless, the results of our proposed ensembles depend on the number of individual classifiers. Note that in some datasets (PageBlocks and PenDigits) the results obtained by *eEnsemble* are better than those obtained using *eClass0*. However, these results change considerably according to the number of individual classifiers. Thus, depending of the data and the environment we should consider which is the best architecture of the corresponding ensemble.

In addition, unlike we could think, the results of the *eEnsembleR* are not significantly better than the obtained by

eEnsemble. And we should take into account that the volume of data received by the individual classifiers of *eEnsembleR* is larger.

Finally, it is very important to remark the importance of the standard deviation in this case. As we can observe, in some ensembles, this value is very variable. However, in many cases this value is very small what indicates that the proposed ensembles are quite stables in those cases.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented and evaluated a proof-of-concept method for constructing ensembles based on individual evolving classifiers. These ensembles (named *eEnsemble* and *eEnsembleR*) are based on evolving classifiers and they keep the properties of the evolving approach classification of streaming data. According to the experimental results, we can conclude that, in general, the results of the proposed ensembles are comparable with those obtained from *eClass0*. However, in some cases, the standard deviation is high what represents that the ensemble created is not very stable.

In order to solve that problem, we could apply some technique (for example, Genetic Algorithms) to decide the best configuration (especially, the value of k) of the ensemble taking into account the nature of the data. Furthermore, evolving systems could be used to evolve the configuration of the ensemble according to the received data. Our main future work is the use of Evolving Systems for creating a stable ensemble which can adapts/evolves its structure on-line and in real time.

ACKNOWLEDGMENT

This work has been supported by the Spanish Government under i-Support (Intelligent Agent Based Driver Decision Support) Project (TRA2011-29454-C03-03).

REFERENCES

- [1] R. Polikar, "Ensemble Based Systems in Decision Making," *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21–45, 2006.
- [2] T. G. Dietterich, "Machine-learning research – four current directions," *AI MAGAZINE*, vol. 18, pp. 97–136, 1997.
- [3] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.
- [4] Y. Freund and R. E. Schapire, "Experiments with a New Boosting Algorithm," in *International Conference on Machine Learning*, 1996, pp. 148–156.
- [5] P. P. Angelov and X. Zhou, "Evolving fuzzy-rule-based classifiers from data streams," *IEEE T. Fuzzy Systems*, vol. 16, no. 6, pp. 1462–1475, 2008.
- [6] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.
- [7] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.
- [8] Q. Zhang and S. Sun, "Evolutionary classifier ensembles for semi-supervised learning," pp. 1–6, Jul. 2010. [Online]. Available: <http://dx.doi.org/10.1109/IJCNN.2010.5596894>

TABLE II: Experimental results: *eClass0* vs. *eEnsembe* and *eEnsembleR*

| Dataset | eClass0 (rate %) | Our approaches - rate % (average) | | | | | |
|------------|----------------------------|--|-------------|--------------------|-------------------|-------------|--------------|
| | | eEnsemble | | | eEnsembleR | | |
| | | <i>k</i> =3 | <i>k</i> =5 | <i>k</i> =10 | <i>k</i> =3 | <i>k</i> =5 | <i>k</i> =10 |
| DataWine | 94,4 | 96,72 ± 2,1 | 95,78 ± 3,3 | 91,11 ± 5,8 | 96,58 ± 1,8 | 95,22 ± 4,1 | 95,44 ± 4,1 |
| CTG | 99,7 | 98,97 ± 1,8 | 97,46 ± 3,4 | 85,40 ± 5,7 | 99,27 ± 0,5 | 99,29 ± 0,5 | 99,26 ± 0,6 |
| Waveform | 80,2 | 78,79 ± 5,3 | 79,36 ± 1,5 | 78,02 ± 5,4 | 79,57 ± 4,5 | 78,51 ± 1,9 | 78,49 ± 2,1 |
| PageBlocks | 69,59 | 76,97 ± 3,3 | 80,99 ± 4,2 | 89,95 ± 2,0 | 71,49 ± 3,0 | 71,80 ± 4,6 | 72,23 ± 4,5 |
| Sat | 61,67 | 61,83 ± 4,2 | 64,40 ± 3,6 | 69,14 ± 2,0 | 59,69 ± 2,5 | 58,85 ± 5,2 | 59,61 ± 5,2 |
| PenDigits | 78,99 | 82,75 ± 1,4 | 81,34 ± 2,0 | 82,06 ± 1,9 | 80,11 ± 2,0 | 79,31 ± 2,1 | 79,31 ± 2,2 |
| Shuttle | 92,98 | 92,89 ± 2,0 | 95,36 ± 1,6 | 88,82 ± 4,6 | 89,95 ± 2,9 | 88,03 ± 4,5 | 88,36 ± 4,3 |

- [9] P. A. D. de Castro, G. P. Coelho, M. F. Caetano, and F. J. V. Zuben, "Designing ensembles of fuzzy classification systems: An immune-inspired approach," in *Artificial Immune Systems: 4th International Conference, ICARIS 2005, Banff, Alberta, Canada, August 14-17, 2005, Proceedings*, ser. Lecture Notes in Computer Science, C. Jacob, M. L. Pilat, P. J. Bentley, and J. Timmis, Eds., vol. 3627. Springer, 2005, pp. 469–482.
- [10] B. V. Dasarathy and B. V. Sheela, "Composite classifier system design: concepts and methodology," in *Proceedings of the IEEE*, vol. 67, 1979, pp. 708–713.
- [11] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 10, pp. 993–1001, Oct. 1990. [Online]. Available: <http://dx.doi.org/10.1109/34.58871>
- [12] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 3, pp. 418–435, May 1992.
- [13] G. Rogova, "Combining the results of several neural network classifiers," *Neural Networks*, vol. 7, no. 5, pp. 777–781, May 1994.
- [14] K. S. Woods, W. P. Kegelmeyer, and K. W. Bowyer, "Combination of multiple classifiers using local accuracy estimates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 405–410, 1997.
- [15] L. I. Kuncheva, J. C. Bezdek, and R. P. W. Duin, "Decision templates for multiple classifier fusion: an experimental comparison," *Pattern Recognition*, vol. 34, pp. 299–314, 2001.
- [16] L. I. Kuncheva, "Classifier ensembles for changing environments," in *Multiple Classifier Systems*. Springer, 2004, pp. 1–15.
- [17] H. G. Ayad and M. S. Kamel, "Cluster-based cumulative ensembles," in *Proceedings of the 6th international conference on Multiple Classifier Systems*, ser. MCS'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 236–245.
- [18] S. Monti, P. Tamayo, J. Mesirov, and T. Golub, "Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data," *Machine Learning*, vol. 52, no. 1-2, pp. 91–118, Jul. 2003.
- [19] C. Cabral, M. Silveira, and P. Figueiredo, "Decoding visual brain states from fmri using an ensemble of classifiers," *Pattern Recognition*, vol. 45, no. 6, pp. 2064 – 2074, 2012, {ce:title}Brain Decoding/{ce:title}. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0031320311001749>
- [20] L. Nanni and A. Lumini, "An experimental comparison of ensemble of classifiers for bankruptcy prediction and credit scoring," *Expert Systems with Applications*, vol. 36, no. 2, Part 2, pp. 3028 – 3033, 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417408000249>
- [21] G. Briem, J. Benediktsson, and J. Sveinsson, "Multiple classifiers applied to multisource remote sensing data," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 40, no. 10, pp. 2291 – 2299, oct 2002.
- [22] A. Amini, M. A. Bagheri, and G. Montazer, "Improving gas identification accuracy of a temperature-modulated gas sensor using an ensemble of classifiers," *Sensors and Actuators B: Chemical*, no. 0, pp. –, 2012.
- [23] P. P. Angelov, *Evolving rule-based models: a tool for design of flexible adaptive systems*. London, UK: Springer-Verlag, 2002.
- [24] P. Angelov and X.-W. Zhou, "Evolving fuzzy systems from data streams in real-time," in *Proceedings of the Internat. Symp. on Evolving Fuzzy Systems*, 2006, pp. 29–35.
- [25] P. Angelov and D. Filev, "An approach to online identification of takagi-sugeno fuzzy models," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 34, no. 1, pp. 484 – 498, feb. 2004.
- [26] X. Zhou and P. Angel, "Real-time joint landmark recognition and classifier generation by an evolving fuzzy system," in *Fuzzy Systems, 2006 IEEE International Conference on*, 2006, pp. 1205 –1212.
- [27] P. P. Angelov and X. Zhou, "Evolving fuzzy classifier for novelty detection and landmark recognition by mobile robots," in *Mobile Robots*, 2007, pp. 89–118.
- [28] P. Angelov, R. Ramezani, and X. Zhou, "Autonomous novelty detection and object tracking in video streams using evolving clustering and takagi-sugeno type neuro-fuzzy system," in *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, june 2008, pp. 1456 –1463.
- [29] Y. Liu, H. Meng, D. Wang, and X. Wang, "Adaptive staggering time estimation for target tracking in periodic nonuniform sampling system," *Electronics Letters*, vol. 43, no. 24, pp. 1385–1387, 2007. [Online]. Available: <http://link.aip.org/link/?ELL/43/1385/1>
- [30] P. Angelov, C. D. Bocaniala, C. Xideas, C. Patchett, D. Ansell, M. Everett, and G. Leng, "A passive approach to autonomous collision detection and avoidance," *Computer Modeling and Simulation, International Conference on*, vol. 0, pp. 64–69, 2008.
- [31] J. A. I. and Javier Ordoez, A. Ledezma, P. de Toledo, and A. Sanchis, "Evolving activity recognition from sensor streams," in *Proceedings of the 2012 IEEE Evolving and Adaptive Intelligent Systems (EAIS-2012)*, May, pp. 96–101.
- [32] J. Ordóñez, J. Iglesias, P. de Toledo, A. Ledezma, A. Sanchis *et al.*, "Online activity recognition using evolving classifiers," *Expert Systems with Applications*, 2012.
- [33] J. A. Iglesias, P. P. Angelov, A. Ledezma, and A. S. de Miguel, "Creating evolving user behavior profiles automatically," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 854–867, 5 2012.
- [34] E. Garca-Cuesta and J. A. Iglesias, "User modeling in changeable environments," in *Proceedings of the 2012 IEEE Evolving and Adaptive Intelligent Systems (EAIS-2012)*, May, pp. 182–185.
- [35] M. Sesmero, J. Alonso-Weber, G. Gutierrez, A. Ledezma, and A. Sanchis, "A new artificial neural network ensemble based on feature selection and class recoding," *Neural Computing and Applications*, vol. 21, pp. 771–783, 2012.
- [36] A. Ledezma, R. Aler, A. Sanchis, and D. Borrajo, "Ga-stacking: Evolutionary stacked generalization," *Intelligent Data Analysis*, vol. 14, no. 1, pp. 89–119, Jan. 2010.
- [37] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, pp. 241–259, 1992.
- [38] R. T. Clemen, "Combining forecasts: A review and annotated bibliography," *International Journal of Forecasting*, vol. 5, no. 4, pp. 559–583, 1989.
- [39] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: <http://archive.ics.uci.edu/ml>